Discrete Optimization 8 (2011) 25–40

Contents lists available at ScienceDirect

Discrete Optimization

journal homepage: www.elsevier.com/locate/disopt

Stable assignment with couples: Parameterized complexity and local search☆

Dániel Marx[a], Ildikó Schlotter[b,*]

[a] Tel Aviv University, Israel

[b] Budapest University of Technology and Economics, Hungary

ARTICLE INFO

Article history:
Received 15 November 2009
Received in revised form 15 July 2010
Accepted 23 July 2010
Available online 6 September 2010

Keywords:
Stable matching
Couples
Local search
Parameterized complexity
Scheduling with job restrictions

ABSTRACT

We study the Hospitals/Residents with Couples problem, a variant of the classical Stable Marriage problem. This is the extension of the Hospitals/Residents problem where residents are allowed to form pairs and submit joint rankings over hospitals. We use the framework of parameterized complexity, considering the number of couples as a parameter. We also apply a local search approach, and examine the possibilities for giving FPT algorithms applicable in this context. Furthermore, we also investigate the matching problem containing couples that is the simplified version of the Hospitals/Residents with Couples problem modeling the case when no preferences are given.

1. Introduction

The classical Hospitals/Residents problem (which is a generalization of the well-known Stable Marriage problem) was introduced by Gale and Shapley [1] to model the following situation. We are given a set of hospitals, each having a number of open positions, and a set of residents applying for jobs in the hospitals. Each resident has a ranking over the hospitals, and conversely, each hospital has a ranking over the residents. Our aim is to assign as many residents to a hospital as possible, with the restrictions that the capacities of the hospitals are not exceeded and the resulting assignment is stable. Stability will be formally defined in Section 3, but essentially an assignment is unstable, if there is a hospital h and a resident r such that r is not assigned to h, but both h and r would benefit from contracting with each other instead of accepting the given assignment.

The original version of the Hospitals/Residents problem is well understood: a stable assignment always exists, and every stable assignment has the same size. (The size of an assignment is the number of residents that have a job.) Moreover, the classical Gale–Shapley algorithm [1] can find a stable assignment in linear time. However, several practical applications motivate some kind of extension or modification of the problem (see e.g. the NRMP program for assigning medical residents in the USA [2,3] or the detailing process of the US Navy [4]), and in the recent decade various versions have been investigated. Among the most frequently studied variants are the case when preference lists involve ties and may be incomplete [5,6], the case when the market of the agents is one-sided, called the Stable Roommates problem [7,8], and the case when the

☆ Supported by ERC Advanced Grant DMMCA and by the Hungarian National Research Fund OTKA 67651.

* Corresponding author.
E-mail address: ildi@cs.bme.hu (I. Schlotter).

1572-5286/$ – see front matter
doi:10.1016/j.disopt.2010.07.004

Table 1
Summary of our results (assuming W[1] $\neq$ FPT).

Task:	Existence problem	Maximum problem	Local search algorithm with FPT running time	
Parameter:	$\|C\|$	$\|C\|$	ℓ	$(\|C\|, \ell)$
Without preferences	P (trivial)	Randomized FPT (Theorem 1)	No permissive alg. (Theorem 7)	Permissive alg. (Theorem 1)
With preferences	No FPT alg. (Theorem 8)	No FPT alg. (Theorem 8)	No permissive alg. (Theorem 10)	Strict alg. (Theorem 11)

assignment may be a many-to-many matching [9–11]. Here we study an extension of this problem, called Hospitals/Residents with Couples, where residents may form couples, and thus have joint rankings over the hospitals. This extension models a situation that arises in many real world applications [3,12], and was first introduced by Roth [2] who also discovered that a stable assignment need not exist when couples are involved. Later, Ronn [13] proved that it is NP-hard to decide whether a stable assignment exists in such a setting. Since then, various approaches have been investigated to deal with the intractability of this problem, but most researchers examined different assumptions on the preferences of couples that guarantee some kind of tractability [14–17].

For the investigation of this problem, we use the framework of parameterized complexity which has been developed by Downey and Fellows [18]. This approach deals with hard problems, where polynomial-time algorithms are unlikely to exist. To do this, we define an integer parameter k for each problem instance, and we try to find algorithms whose running time remains tractable if the parameter k is small. More precisely, we look for algorithms whose running time is of the form $f(k)n^c$, where n is the size of the instance having parameter k, f is an arbitrary function, and c is a constant. Note that the running time may depend exponentially or worse on the parameter k, but it yields a polynomial of degree c for each fixed k. Problems admitting such an algorithm are called fixed-parameter tractable or FPT.

Up to our knowledge, no version of the Hospitals/Residents problem has been studied from the parameterized point of view. When considering Hospitals/Residents with Couples, the number of couples in an instance can be a natural parameter. We prove the negative result of Theorem 8 stating that deciding the existence of a stable assignment for the Hospitals/Residents with Couples problem is not FPT, provided that W[1] $\neq$ FPT holds (which is a standard assumption of parameterized complexity theory).

Local search is a basic technique that has been widely applied in heuristics for practical optimization problems for several decades [19]. However, investigations considering the connection of local search and parameterized algorithms have only been started a few years ago, and research in this area has been gaining increasing attention lately [20]. The basic idea of local search is to find an optimal solution by an iteration in which we improve the current solution step by step through local modifications. Local search can become more efficient if we can decide whether there exists a better solution S' that is ℓ modification steps away from a given solution S. Typically, the ℓ-neighborhood of a solution S can be explored in $n^{O(\ell)}$ time by examining all possibilities to find those parts of S that should be modified. (Here n is the input size.) However, in some cases the dependency on ℓ can be improved by getting ℓ out of the exponent of n, resulting in a running time of the form $f(\ell)n^c$ for some constant c, meaning that the neighborhood exploration problem is FPT. This question has already been studied in connection with different optimization problems [21–23].

In Theorem 11, we give an algorithm for the following problem: given a stable assignment S, find a stable assignment S' of greater size which can be obtained from S by modifying the assignment for at most ℓ residents. The presented algorithm guesses the structure of the modification needed to obtain the larger stable assignment S', and applies color-coding to localize this structure step by step in the original instance, using only simple steps. The running time of this algorithm for an instance of size n involving a set C of couples is $f(\ell, |C|)n^c$ for some function f and constant c, so this yields an FPT algorithm with parameters ℓ and $|C|$. In contrast, if we only regard ℓ as a parameter, then Theorem 10 shows that no FPT algorithm exists for this problem unless $W[1] = \text{FPT}$.

We also contribute to the framework of parameterized local search algorithms by distinguishing between "strict" algorithms that perform the local search step in some neighborhood of a solution as described above, and "permissive" algorithms whose task is the following: given some problem with an initial solution S, find *any* better solution, provided that a better solution exists in the local neighborhood of S. Our motivation for this distinction is that finding an improved solution in the neighborhood of a given solution may be hard, even for problems where an optimal solution is easily found. We hope that this differentiation clarifies the role of local search in such cases, helping the parameterized complexity analysis of such problems.

Most of the questions examined here are also worth studying in a model that does not involve preferences. This simplification leads to a matching problem that we call Maximum Matching with Couples. Using a result by the first author concerning matroids from the parameterized view-point [24], we propose a randomized algorithm in Theorem 1 that finds a matching of maximum size, and runs in FPT time if the parameter is the number $|C|$ of couples. Therefore, this problem becomes easier without preferences. By contrast, the local search problem still remains hard to solve (Theorem 7). For a summary of our results see Table 1.

The paper is organized as follows. Section 2 covers our notation and the preliminaries, and Section 3 introduces the formal definitions of the problems examined in the paper. In Section 4, we investigate the Matching with Couples problem, and we present our results on the Hospitals/Residents with Couples problem in Section 5. Finally, we give a short summary of our results in Section 6.

2. Preliminaries

For some integer k, we use $[k] = \{1, 2, \ldots, k\}$, and $\binom{[k]}{2} = \{(i, j) \mid 1 \leq i < j \leq k\}$. For a graph G, $V(G)$ denotes its vertices and $E(G)$ its edges. A *matching* in G is a set M of edges such that no two edges in M share an endpoint. If x is an endpoint of some edge in M, then x is *covered* by M. For some x covered by M, $M(x) = y$ if $xy \in M$.

A *decision problem* Q over some alphabet Σ is an arbitrary subset of Σ^*, and an algorithm *decides* Q if for every $x \in \Sigma^*$, its output is 'Yes' if and only if $x \in Q$. With each instance of an *optimization problem* Q we associate a set of solutions and an objective function which we want to maximize or minimize. In this paper we consider only maximization problems.

Parameterized complexity. A *parameterized problem* is a pair (Q, κ) where $Q \subseteq \Sigma^*$ is a decision problem over some alphabet Σ, and $\kappa : \Sigma^* \to \mathbb{N}$ is a *parameterization* of the problem, assigning a *parameter* to each instance of Q. An algorithm is *fixed-parameter tractable* or FPT, if its running time is $f(k)|I|^c$ for some computable function f, where I is the input, k is the parameter and c is a constant. A parameterized problem is FPT, if there is an FPT algorithm that decides it. Analogously to the classical complexity theory, the theory of $W[1]$-*hardness* can be used to prove that some problem is not FPT, unless the widely believed FPT $\subset$ W[1] conjecture fails. Given two parameterized problems (Q_1, κ_1) and (Q_2, κ_2) over the alphabet Σ, an *FPT reduction* from (Q_1, κ_1) to (Q_2, κ_2) is a function $g : \Sigma^* \to \Sigma^*$, computable by an FPT algorithm, such that $I \in Q_1$ if and only if $g(I) \in Q_2$ and $\kappa_2(g(I)) \leq f(\kappa_1(I))$ for some computable function f, for every $I \in \Sigma^*$. To prove $W[1]$-hardness results, we use that the class of $W[1]$-hard problems is closed under FPT reductions. The FPT reductions in this paper are from the $W[1]$-hard parameterized problem Clique, in which a graph G and a parameter k is given, and the task is to decide whether there is a clique of size k in G. For further details on parameterized complexity, see e.g. [18,25], or [26].

Local search. To formalize the task of a local search algorithm, let Q be an optimization problem with an objective function T which we want to maximize. To define the concept of neighborhoods, we suppose there is some *distance* $d(x, y)$ defined for each pair (x, y) of solutions for some instance I of Q. We say that x is ℓ-*close* to y if $d(x, y) \leq \ell$. The input of a local search algorithm for Q is an instance I of Q, a solution S_0 for I, and an integer ℓ. A *strict local search* algorithm for Q has the following task:

> Strict local search for Q
> Input: (I, S_0, ℓ) where I is an instance of Q, S_0 is a solution for I, and $\ell \in \mathbb{N}$.
> Task: If there exists a solution S for I such that $d(S, S_0) \leq \ell$ and $T(S) > T(S_0)$, then output such an S.

In contrast, a *permissive local search* algorithm for Q is allowed to output a solution that is not close to S_0, provided that it is better than S_0. In local search methods, such an algorithm is as useful as its strict version. Formally, its task is as follows:

> Permissive local search for Q
> Input: (I, S_0, ℓ) where I is an instance of Q, S_0 is a solution for I, and $\ell \in \mathbb{N}$.
> Task: If there exists a solution S for I such that $d(S, S_0) \leq \ell$ and $T(S) > T(S_0)$, then output *any* solution S' for I with $T(S') > T(S)$.

On the one hand, note that if an optimal solution can be found by some algorithm, then this yields a permissive local search algorithm for the given problem. On the other hand, finding a strict local search algorithm might be hard even if an optimal solution is easily found. An example for such a case is the Minimum Vertex Cover problem for bipartite graphs [22]. Besides, proving that no permissive local search algorithm exists for some problem is clearly more relevant than proving that no strict local search algorithm exists for it (having a certain running time). We also present results of this kind.

We remark that the distinction between permissive and strict local search algorithms cannot be maintained when addressing the standard decision version of these problems. To see this, consider the following formulation of such a local search problem: given an instance I of some optimization problem Q, a solution S_0 to I, and some $\ell \in \mathbb{N}$, decide whether there is a solution S for I such that $d(S, S_0) \leq \ell$ and $T(S) > T(S_0)$. Clearly, the difference between the strict and the permissive approach is no longer applicable in this definition. Consequently, instead of proving $W[1]$-hardness for problems considering local search algorithms, our hardness results will be formulated as statements that no (permissive) local search algorithm can run in FPT time with a certain parameterization, assuming FPT $\neq$ W[1].

3. Problem definitions

In this section we give the formal descriptions of the different models that we investigate.

Model without preferences. First, we define two versions of the Hospitals/Residents problem that involve couples, but do not deal with preferences, using only a notion of acceptability instead.

A *couples' market with acceptance*, or *cma* for short, consists of a set S of singles, a set C of couples, a set H of hospitals together with a *capacity* $f(h)$ for each hospital h, a set $A(s) \subseteq H$ for each single $s \in S$ representing *acceptable hospitals* for s, and a set $A(c) \subseteq \widetilde{H}$ for each couple $c \in C$ representing *acceptable hospital pairs* for c. Here $\widetilde{H} = (H \cup \{\oslash\}) \times (H \cup \{\oslash\}) \setminus \{(\oslash, \oslash)\}$ where $\oslash$ is a special symbol indicating that someone is unemployed. If $f \equiv f_0$ for some $f_0 \in \mathbb{N}$, then we say that the cma is f_0*-uniform*. Each couple c is a pair $(c(1), c(2))$, and we call the elements of the set $R = \bigcup_{c \in C}\{c(1), c(2)\} \cup S$ *residents*. The couples are mutually disjoint, i.e. each resident appears in at most one couple.

An *assignment* for a cma (S, C, H, f, A) is a function $M : R \to H \cup \{\oslash\}$ such that $M(s) \in A(s) \cup \{\oslash\}$ for each $s \in S$, $M(c) \in A(c) \cup \{(\oslash, \oslash)\}$ for each $c \in C$, and the number of residents assigned to a hospital h is at most its capacity $f(h)$. Here, $M(c)$ denotes the pair $(M(a), M(b))$ for some couple $c = (a, b)$, and the set of residents assigned to h in M is the set $\{r \mid r \in R, M(r) = h\}$, denoted by $M(h)$. We say that an assignment M *covers* a resident r if $M(r) \neq \oslash$, and M covers a couple c, if it covers $c(1)$ or $c(2)$. We define the *size* of M, denoted by $|M|$, to be the number of residents covered by M. The Maximum Matching with Couples problem is an optimization problem, where given a cma I, the set of solutions is the set of assignments for I, and the task is to find an assignment for I of maximum size.

To consider the local search versions of the Maximum Matching with Couples problem, we define the *distance* $d(M, M')$ of two assignments M and M' for some cma I as the number of residents r for which $M(r) \neq M'(r)$. Using this, the task of a strict local search algorithm for Maximum Matching with Couples is the following: given a cma I together with an assignment M for I, and some integer ℓ, find an assignment M' for I with $d(M, M') \leq \ell$ that has size greater than M. The input of a permissive local search algorithm for Maximum Matching with Couples is the same, but in this case the task is to find any assignment for I having size greater than M, if such an assignment exists in the ℓ-neighborhood of M.

Model with preferences. Next, let us define some versions of the Hospitals/Residents problem, where couples are involved and preferences play an important role.

A *couples' market with preference*, or *cmp* for short, consists of the sets S, C, and H representing singles, couples, and hospitals, respectively, a capacity $f(h)$ for each $h \in H$, and a *preference list* $L(a)$ for each $a \in S \cup C \cup H$. The set $A = S \cup C \cup H$ is called the set of *agents*. The preference lists can be incomplete, but cannot involve ties, so if $s \in S$ then $L(s)$ is a strictly ordered set of hospitals, if $c \in C$ then $L(c)$ is a strictly ordered subset of $\widetilde{H}$, and if $h \in H$ then $L(h)$ is a strictly ordered set of residents. Here, $\widetilde{H}$ and the symbol $\oslash$ are defined the same way as for the case without preferences, and we also adopt the notion of f_0-uniformity. The set of elements appearing in the list $L(a)$ is $A^L(a)$, and we say that x is *acceptable* for a if $x \in A^L(a)$. Clearly, we may assume that acceptance is mutual, so $h \in A^L(s)$ holds if and only if $s \in A^L(h)$ for each $s \in S$ and $h \in H$, and $(h_1, h_2) \in A^L(c)$ implies $c(i) \in A^L(h_i)$ or $h_i = \oslash$ for both $i \in \{1, 2\}$, for each $c \in C$. For some $x \in A^L(a)$, the *rank* of x w.r.t. a, denoted by $\rho(a, x)$, is $r \in \mathbb{N}$ if x is the r-th element in $L(a)$. If $x \notin A^L(a)$, then we let $\rho(a, x) = \infty$ for all meaningful x.

Let $I = (S, C, H, f, L)$ be a cmp. An *assignment* for I is an assignment for the underlying cma (S, C, H, f, A^L). We say that x is *beneficial* for the agent a with respect to an assignment M if $x \in A^L(a)$ and one of the following cases holds: (1) $a \in S \cup C$ and either a is not covered by M or $\rho(a, x) < \rho(a, M(a))$, (2) $a \in H$ and either $|M(a)| < f(a)$ or there exists a resident $r' \in M(h)$ such that $\rho(a, x) < \rho(a, r')$. A *blocking pair* for M can be of three types:

- it is either a pair formed by a single s and a hospital h such that both s and h are beneficial for each other w.r.t. M,
- or a pair formed by a couple c and a pair (h_1, h_2) with $h_1 \neq h_2$ such that (h_1, h_2) is beneficial for c w.r.t. M, and for both $i \in \{1, 2\}$ it holds that if $h_i \neq \oslash$ then either $c(i)$ is beneficial for h_i w.r.t. M or $c(i) \in M(h_i)$,
- or a pair formed by a couple c and a hospital h such that (h, h) is beneficial for c w.r.t. M, and *the couple c is beneficial for h*. If h prefers $c(1)$ to $c(2)$, this latter means that either $|M(h)| \leq f(h) - 2$, or $|M(h)| \leq f(h) - 1$ and $\rho(h, c(1)) < \rho(h, r)$ for some $r \in M(h)$, or $\rho(h, c(1)) < \rho(h, r_1)$ and $\rho(h, c(2)) < \rho(h, r_2)$ for some $r_1 \neq r_2$ in $M(h)$.[1]

An assignment M for I is *stable* if there is no blocking pair for M.

The input of the Hospitals/Residents with Couples problem is a cmp I, and the task is to determine a stable assignment for I, if such an assignment exists. We denote by Maximum Hospitals/Residents with Couples the optimization problem where the task is to determine a stable assignment of maximum size for a given cmp. Another variant of this optimization problem which we will address is the Increase Hospitals/Residents with Couples problem. Here, the input is a cmp I and a stable assignment M_0 for I, and the task is to find a stable assignment with size at least $|M_0| + 1$.

4. Matching without preferences

In this section we investigate the parameterized complexity of the Maximum Matching with Couples problem. In Section 4.1 we present a randomized FPT algorithm for it, where the parameter is the number of couples. Then we turn our attention to an application of this algorithm in the context of scheduling problems in Section 4.2. We also examine the possibility of finding a local search algorithm for the Maximum Matching with Couples problem in Section 4.3.

4.1. Fixed-parameter tractability

Let us examine the complexity of the Maximum Matching with Couples problem. Clearly, if there are no couples in a given instance, then the problem is equivalent to finding a maximum matching in a bipartite graph, and can be solved by standard techniques. If couples are involved, the problem becomes hard. More precisely, the decision version of this problem is NP-complete [27,28], even in the special case where each hospital has a capacity of 2, and the acceptable hospital pairs for a couple are always of the form (h, h) for some $h \in H$. However, if the number of couples is small, which is a reasonable assumption in many practical applications, Maximum Matching with Couples becomes tractable, as shown by Theorem 1.

1 We thank David Manlove for pointing out this case.

Theorem 1. Maximum Matching with Couples *can be solved in randomized FPT time with parameter* $|C|$.

We will use the following lemma to solve a special case of the Maximum Matching with Couples problem.

Lemma 2. *There exists an algorithm running in randomized FPT time with parameter* $|C|$ *that, given an instance of the* 1*-uniform* Maximum Matching with Couples *problem and some integer* n*, finds an assignment covering at least* n *singles and also each resident that is a member of some couple, if such an assignment exists.*

To prove Lemma 2, we need some results from [24] concerning matroids.

Although we only use basic concepts from matroid theory, here we give a brief outline of the main definitions used. For some set U and collection $\mathcal{I} \subseteq 2^U$, the pair $(U, \mathcal{I})$ is a *matroid* if the following hold: (1) $\emptyset \in \mathcal{I}$, (2) if $X \in \mathcal{I}$ and $X' \subseteq X$ then $X' \in \mathcal{I}$, and (3) if $X, Y \in \mathcal{I}$ and $|X| < |Y|$ then $X \cup \{y\} \in \mathcal{I}$ for some $y \in Y \setminus X$. The elements of $\mathcal{I}$ are called *independent sets*. A matrix A over a field F is a *linear representation* of a matroid $(\{u_i \mid i \in [n]\}, \mathcal{I})$, if for any set J of indices in $[n]$, the set of columns in A corresponding to the indices J are independent over F if and only if $\{u_j \mid j \in J\} \in \mathcal{I}$. A matroid is *linear* if it admits a linear representation. A maximal independent set of a matroid is called a *basis* of the matroid. The *dual* of a matroid $(U, \mathcal{I})$ with basis set $\mathcal{B}$ is the matroid with ground set U whose basis set is $\{U \setminus B \mid B \in \mathcal{B}\}$. The *k-truncation* of $(U, \mathcal{I})$ is the matroid $(U, \mathcal{I}')$ where $I \in \mathcal{I}'$ if and only if $I \in \mathcal{I}$ and $|I| \le k$. Given a bipartite graph $G(A, B; E)$, its *transversal matroid* has ground set A, and X is defined to be independent if there is a matching in G covering X.

Theorem 3 (*[24]*)**.** *Let* $\mathcal{M}(U, \mathcal{I})$ *be a linear matroid where the ground set* U *is partitioned into blocks of size* b*. Given a linear representation* A *of* $\mathcal{M}$*, it can be determined in* $f(k, b) \cdot \|A\|^{O(1)}$ *randomized time whether there is an independent set that is the union of* k *blocks. (*$\|A\|$ *denotes the length of* A *in the input.)*

The following generalization of Theorem 3 will be convenient for our purposes.

Corollary 4. *Let* $\mathcal{M}(U, \mathcal{I})$ *be a linear matroid and let* $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$ *be a collection of subsets of* U*, each of size* b*. Given a linear representation* A *of* $\mathcal{M}$*, it can be determined in* $f(k, b) \cdot \|A\|^{O(1)}$ *randomized time whether there is an independent set that is the union of* k *disjoint sets in* $\mathcal{X}$*.*

Proof. First, let us make $n(u)$ copies for each $u \in U$, where $n(u)$ is the number of sets in $\mathcal{X}$ containing u, i.e. let $U' = \{u^i \mid u \in U, n(u) > 0, i \in [n(u)]\}$. Let $\mathcal{M}'(U', \mathcal{I}')$ be the matroid where $\mathcal{I}'$ contains those sets which can be obtained from some set $I \in \mathcal{I}$ by replacing each $u \in I$ with an arbitrary element from $\{u^i \mid i \in [n(u)]\}$. A representation A' of $\mathcal{M}'$ can be obtained from A by putting $n(u)$ copies of the column representing u into A' for each $u \in U$. For each $i \in [n]$, let $X'_i \subseteq U'$ be obtained by replacing each element u in X_i with u^j if X_i is the j-th set in $\mathcal{X}$ containing u. Clearly, by letting X'_i be a block (having size b) for each $i \in [n]$, we get a partition of U'.

The sets $\{X_{i_j} \mid j \in [k]\}$ satisfy the requirements (being disjoint and having an independent union in $\mathcal{M}$) if and only if the sets $\{X'_{i_j} \mid j \in [k]\}$ are k blocks whose union is independent in $\mathcal{M}'$, and thus the algorithm of Theorem 3 provides the solution. □

Lemma 5 (*[24]*)**.** (1) *Given a representation* A *over a field* F *of a matroid* $\mathcal{M}$*, a representation of the dual matroid* $\mathcal{M}^*$ *over* F *can be found in polynomial time.*
(2) *Given a representation* A *over* $\mathbb{N}$ *of a matroid* $\mathcal{M}$ *and an integer* k*, a representation of the k-truncation of* $\mathcal{M}^k$ *can be found in randomized polynomial time.*
(3) *Given a bipartite graph* $G(A, B; E)$*, a representation of its transversal matroid over* $\mathbb{N}$ *can be constructed in randomized polynomial time.*

Now, we are ready to prove Lemma 2.

Proof (of Lemma 2). Let (S, C, H, f, A) be the given cma with $f \equiv 1$ for which we have to find an assignment covering at least n singles and each resident that is a member of some couple in C. Clearly, we can assume $A(c) \subseteq H \times H$.

Let $G(H, S; E)$ be the bipartite graph where a single $s \in S$ is connected with a hospital $h \in H$ if and only if $h \in A(s)$. We can assume w.l.o.g. that G has a matching of size at least n as otherwise no solution may exist, and this case can be detected easily in polynomial time. We define $\mathcal{M}(H, \mathcal{I})$ to be the matroid where a set $X \subseteq H$ is independent if and only if there is a matching in G that covers at least n singles but covers no hospitals from X. Observe that $\mathcal{M}$ is exactly the dual of the n-truncation of the transversal matroid of G, and thus it is indeed a matroid. By Lemma 5, we can find a linear representation A of $\mathcal{M}$ in randomized polynomial time.

We define the matroid $\mathcal{M}'(U, \mathcal{I}')$ with ground set $U = H \cup C$ such that $X \subseteq U$ is independent in $\mathcal{M}'$ if $X \cap H$ is independent in $\mathcal{M}$. A representation of $\mathcal{M}'$ can be obtained by appending a unit matrix of size $k \times k$ to A in the intersection of k newly introduced rows and columns, each containing only zeros in the remaining entries. Let $\mathcal{X}$ be the collection of the sets that are of the form $\{c, h_1, h_2\}$ where $c \in C$ and $(h_1, h_2) \in A(c)$.

Observe that if $X_1, X_2, \ldots, X_k$ are k disjoint sets in $\mathcal{X}$ whose union is independent in $\mathcal{M}'$, then we can construct an assignment covering each resident that is a member of some couple and at least n additional singles as follows. For each $\{c, h_1, h_2\} \in \{X_1, \ldots, X_k\}$ we choose $M(c)$ from $\{(h_1, h_2), (h_2, h_1)\} \cap A(c)$ arbitrarily. The disjointness of the sets $X_1, \ldots, X_k$

guarantees that this way we assign exactly one resident to each hospital in $X = \bigcup_{i\in[k]} X_i \cap H$. Now, let N be a matching in G that covers at least n singles, but no hospitals from X. Such a matching exists, as X is independent in $\mathcal{M}$. Thus, letting $M(s)$ be $N(s)$ if s is covered by N and $\oslash$ otherwise for each $s \in S$ yields that M is an assignment with the desired properties. Conversely, if M is an assignment covering each member of the couples and n additional singles, then the sets $\{c, h_1, h_2\}$ for each $c \in C$ and $M(c) = (h_1, h_2)$ form a collection of k disjoint sets in $\mathcal{X}$ whose union is independent in $\mathcal{M}'$. By Corollary 4, such a collection can be found in randomized FPT time when k is the parameter, yielding a solution if it exists. □

Using Lemma 2, we can prove Theorem 1.

Proof. Let $I = (S, C, H, f, A)$ be the given cma in the Maximum Matching with Couples problem. We give an algorithm that decides whether there is an assignment covering t residents in this instance.

First, we reduce the general problem to the 1-uniform case by "cloning" the hospitals. To this end, substitute each $h \in H$ with newly introduced hospitals $h^1, \dots, h^{f(h)}$; the set of acceptable residents will be $A(h)$ for each of these hospitals. Now, for each single s and for each hospital h acceptable for s, replace h with the elements $h^1, \dots, h^{f(h)}$ in the set of acceptable hospitals for s. Also, for each couple c and for each entry (h_a, h_b) in $A(c)$, replace the entry (h_a, h_b) in the set of of acceptable hospital pairs for c with the elements in $\{(h_a^i, h_b^j) \mid i \in [f(h_a)], j \in [f(h_b)]\}$. (The cases where $h_a = \oslash$ or $h_b = \oslash$ can be handled similarly.)

It is easy to see that an assignment for I covering a certain set of residents can be transformed into an assignment for the modified instance covering exactly the same residents, and vice versa. Note that this modification increases the input length of the instance by at most a factor of $f_{\max}^2$, where $f_{\max}$ is the maximum capacity of some hospital in H. Since we can assume $f_{\max} \leq |S| + 2|C|$ without losing generality, this means that the input increases only by a polynomial factor.

Next, we show how to solve the 1-uniform Maximum Matching with Couples problem using the algorithm of Lemma 2. For each couple $c \in C$, we branch into three cases, according to the cases where want to cover 0, 1, or 2 of the residents of the couple c. In the branch where we do not want to cover any member of c, we simply delete c from the market. In the branch where we only want to cover one member of the couple c, we can replace c with a new single s_c that finds exactly those hospitals h acceptable for which either $(\oslash, h)$ or $(h, \oslash)$ was acceptable for c. (We also have to replace the members of c with s_c in the acceptance lists of the hospitals.) After branching for each couple in C, we look for an assignment that covers each resident in the remaining set C' of couples, and also $t - 2|C'|$ additional singles. This task can be accomplished by using the algorithm of Lemma 2. Notice that the branchings only increase the running time by a factor of $3^{|C|}$.

Clearly, such an assignment yields an assignment of size t in the original instance I. Conversely, if there is an assignment of size t in I, then at least one branch will lead to such an assignment. □

We remark that the main obstacle to derandomize the algorithm of Theorem 1 is the fact that the proof of Theorem 3 makes use of the Zippel–Schwartz Lemma in some issues connected to matroid representations, and hence is inherently randomized (see also [24]).

4.2. *An application in scheduling*

Lemma 2 can be generalized in a straightforward way to the case when there are *groups* having some fixed size $p \in \mathbb{N}$ instead of couples of size 2 in the given market. Then, in the proof of Lemma 2 we have to use blocks of size $p + 1$ instead of size 3, and hence we need to apply Corollary 4 with setting $b = p + 1$. The running time of this more general version of the algorithm is randomized FPT, if we regard both the number of groups and p as parameters. This has a useful consequence in connection to the following scheduling problem.

We are given a set $\mathcal{M}$ of parallel *machines* and a set J of independent *jobs*. With each job $j \in J$ we associate a *processing time* p_j and a *processing set* $\mathcal{M}_j \subseteq \mathcal{M}$ of machines that can process j. The task is to find a scheduling where each job j is (entirely) processed by a machine in $\mathcal{M}_j$. Formally, a *scheduling* in this setting is an assignment $\mu : J \to \mathcal{M}$ mapping each job $j \in J$ to some machine in its processing set $\mathcal{M}_j$. The *makespan* of a scheduling μ is the value $\max_{m\in\mathcal{M}} \sum_{\mu(j)=m} p_j$, which describes the latest completion time when some machine in $\mathcal{M}$ finishes all the jobs assigned to it.

We consider the following problem in this context: given the set of jobs with their processing times and processing sets, find a scheduling that minimizes the makespan. In the standard three-field notation of the area of scheduling algorithms, this problem is abbreviated as $P|\mathcal{M}_j|C_{\max}$.

Similar scheduling problems have been widely studied by researchers, see the recent survey of Leung and Li [29]. Due to the computational hardness of these problems, most of the work in this area focuses on either approximation or exponential-time algorithms. Also, researchers have extensively studied special cases which are more likely to be tractable, such as scheduling with unit length jobs, or cases where the job processing restrictions exhibit some specific structure.

Here we complement this line of research by providing a randomized fixed-parameter tractable algorithm for the special case of the $P|\mathcal{M}_j|C_{\max}$ problem where k jobs have processing time $p \in \mathbb{N}$ and all other jobs have processing time 1, and we regard k as a parameter. This problem was proved to be NP-complete even if $p = 2$ (see [27,28]), so investigating the parameterized complexity of this problem might be of practical importance.

Theorem 6. *There is a randomized FPT algorithm for the special case of the $P|\mathcal{M}_j|C_{\max}$ problem where k jobs have processing time $p \in \mathbb{N}$ and all other jobs have processing time 1, and we regard k and p as the parameter.*

Proof. Let us be given some instance I of the $P|\mathcal{M}_j|C_{\max}$ problem where k jobs have processing time $p \in \mathbb{N}$ and all other jobs have processing time 1. Let J and $\mathcal{M}$ denote the set of jobs and machines, and let p_j and $\mathcal{M}_j$ denote the processing time and the processing set for some job $j \in J$, respectively.

To construct a minimum makespan scheduling, we will use the generalized version of the algorithm presented in Lemma 2, dealing with case when there are groups of some fixed size p instead of couples having size 2 in the market. This algorithm runs in randomized FPT time, if both k and p are parameters.

We provide an algorithm that for any $T \in \mathbb{N}$ can construct a scheduling for the given instance I with makespan at most T, if such a scheduling exists. By applying a binary search on the value of T we can extend this to finding a minimum makespan scheduling. For a given value $T \in \mathbb{N}$, we construct a 1-uniform cma as follows. Let $z = \lfloor T/p \rfloor$. For each machine $m \in \mathcal{M}$ we define T hospitals $h_m^1, h_m^2, \ldots, h_m^T$ (each with capacity 1). For each job j with processing time 1, we add a corresponding single resident s_j that finds exactly the hospitals in $\{h_m^i \mid m \in \mathcal{M}_j, i \in [T]\}$ acceptable. For each job j with processing time p, we add a corresponding group g_j consisting of p newly introduced residents, and we let the acceptable p-tuple of hospitals for g_j be $\{(h_m^{(i-1)p+1}, h_m^{(i-1)p+2}, \ldots, h_m^{ip}) \mid m \in \mathcal{M}_j, i \in [z]\}$. We apply the generalized version of the algorithm of Lemma 2 to find an assignment covering each resident in this instance.

Note that the number of groups is k (the number of jobs with processing time p) and the number of singles is $|J| - k$ (the number of remaining jobs). It is also easy to see that the construction time is polynomial in T and the original input length, observe that $T \leq p|J|$ can also be assumed. Thus, the presented algorithm runs in randomized FPT time with parameters k and p.

It remains to show the correctness of the algorithm. First we prove that any assignment covering each resident can be used to construct a scheduling μ for I with makespan at most T as follows. If some single s_j is assigned to some hospital in h_m^i, then we let $\mu(j) = m$; note that $\mu(j) \in \mathcal{M}_j$ holds. Similarly, for each group g_j assigned to a p-tuple of hospitals $(h_m^{(i-1)p+1}, \ldots, h_m^{ip})$ for some $m \in \mathcal{M}_j$ and $i \in [z]$, we let $\mu(j) = m$. Clearly, this is a scheduling for I with makespan at most T.

For the other direction, suppose that μ is a scheduling with makespan T. Let J_m be the set of jobs assigned to some machine m, and suppose that J_m contains the jobs $j_{x_1}, \ldots, j_{x_a}$ with processing time p, and the jobs $j_{y_1}, \ldots, j_{y_b}$ with processing time 1. In this case, we assign each group g_{x_i} to the p-tuple of hospitals $(h_m^{(i-1)p+1}, \ldots, h_m^{ip})$, and we assign each single s_{y_i} to the hospital h_m^{pa+i}. It is easy to verify that this way we indeed obtain an assignment in the constructed cma covering each resident. $\square$

4.3. Local search

Here, we investigate the applicability of the local search approach to handle the intractability of the Maximum Matching with Couples problem.

Theorem 7. *There is no permissive local search algorithm for the* 2*-uniform* Maximum Matching with Couples *that runs in FPT time with parameter ℓ denoting the radius of the explored neighborhood, unless* W[1] = FPT.

Proof. Let G be the input graph for the Clique problem and k be the parameter given. We denote the vertices of G by $v_1, v_2, \ldots, v_n$. We claim that if there is a permissive local search algorithm $\mathcal{A}$ for Maximum Matching with Couples running in FPT time with parameter ℓ, then we can use $\mathcal{A}$ to solve Clique in FPT time. To prove this, we construct an input $\Lambda = (I, M_0, \ell)$ of $\mathcal{A}$ with the following properties: every assignment for I with size at least $|M_0| + 1$ is ℓ-close to M_0, and there is such an assignment for I if and only if G has a clique of size k. Thus, G has a clique of size k if and only if $\mathcal{A}$ outputs an assignment for I with size at least $|M_0| + 1$.

To construct Λ, we first define the cma I together with the assignment M_0 for it. Let the set H of hospitals be the union of $D = B \cup \bigcup\{H^{i,j} \mid i, j \in [k]\}$, $D' = B' \cup \bigcup\{H'^{i,j} \mid i, j \in [k]\}$ and $F = \{f_i \mid i \in [k]\}$, where $B = \{b_i \mid i \in [2k-1]\}$, $H^{i,i} = \{h_{j,j}^{i,i} \mid j \in [n]\}$ for each $i \in [k]$, $H^{i,j} = \{h_{x,y}^{i,j} \mid v_x v_y \in E(G)\}$ for each $i \neq j$, $\{i, j\} \subseteq [k]$, and for each hospital h in B ($H^{i,j}$, respectively) we also define a hospital h' to be in B' ($H'^{i,j}$, respectively). For brevity, we will use the notation $H_{h,\bullet}^{i,j} = \{h \mid \exists y : h = h_{h,y}^{i,j} \in H^{i,j}\}$ and $H_{\bullet,h}^{i,j} = \{h \mid \exists x : h = h_{x,h}^{i,j} \in H^{i,j}\}$. The capacity of each hospital is 2. For each hospital $h \in D$ we define a couple denoted by $c(h)$, and for each $h' \in D'$ we define two singles $s_1(h')$ and $s_2(h')$. Let $C = \{c(h) \mid h \in D\}$ and let $S = \{s_0\} \cup \{s_i(h') \mid h' \in D', i \in \{1, 2\}\}$.

Before defining $A(p)$ for each $p \in S \cup C$, we define the assignment M_0 for I, as this will not cause any confusion. Let $M_0(s_0) = \emptyset$, and let $M_0(p) = h$ where either $h \in D$ and p is a member of the couple $c(h)$, or $h \in D'$ and $p \in \{s_1(h), s_2(h)\}$. Now, for each $p \in S \cup C$, we define the set of acceptable hospitals or pairs of hospitals $A(p)$ to be the union of $\{M_0(p)\}$ and the set $A'(p)$ of hospitals, defined below, that can be assigned to p besides $M_0(p)$. We define $A'(p)$ for each $p \in S \cup C$ as follows.

$A'(c(h)) = \{(h', h')\}$ for each $h \in D$
$A'(s_0) = \{b_1\}$
$A'(s_1(b_i')) = H^{1,i}$ for each $i \in [k]$
$A'(s_2(b_i')) = \{b_{i+1}\}$ for each $i \in [k]$
$A'(s_1(b_{k+i}')) = H^{i,1}$ for each $i \in [k-1]$

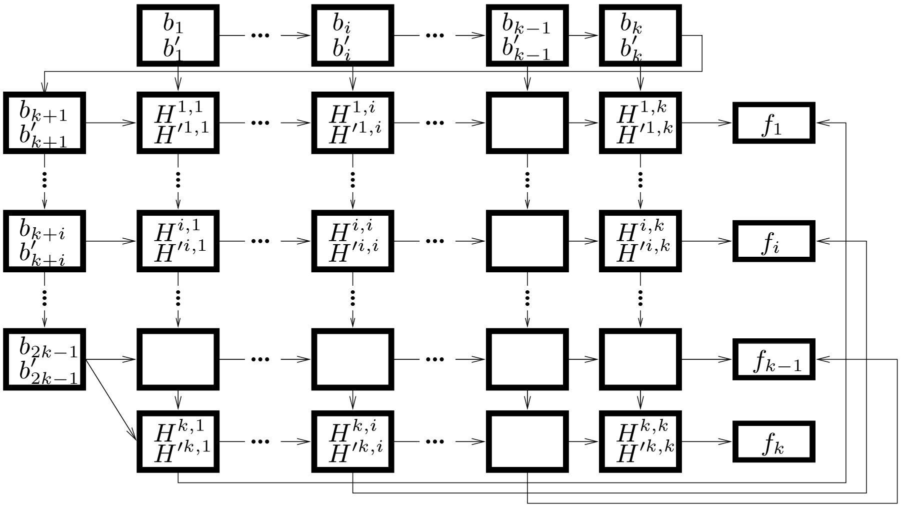

Fig. 1. A block diagram showing the hospitals in the proof of Theorem 7. For two sets H_1, H_2 of hospitals, (H_1, H_2) is an arc if $A'(s) \subseteq H_2$ for some $s \in S$ with $M_0(s) \in H_1$.

$A'(s_2(b'_{k+i})) = \{b_{k+i+1}\}$ for each $i \in [k-2]$
$A'(s_2(b'_{2k-1})) = H^{k,1}$
$A'(s_1(h'^{i,j}_{x,y})) = H^{i,j+1}_{x,\bullet}$ for each $i \in [k], j \in [k-1]$ and every possible x and y
$A'(s_1(h'^{i,k}_{x,y})) = \{f_i\}$ for each $i \in [k]$ and every possible x and y
$A'(s_2(h'^{i,j}_{x,y})) = H^{i+1,j}_{\bullet,y}$ for each $i \in [k-1], j \in [k]$ and every possible x and y
$A'(s_2(h'^{k,i}_{x,y})) = \{f_i\}$ for each $i \in [k]$ and every possible x and y.

This completes the definition of the cma $I = (S, C, H, f, A)$. Observe that M_0 indeed is an assignment for I. Finally, setting $\ell = 4k^2 + 8k - 3$ finishes the definition of the instance $\Lambda = (I, M_0, \ell)$. Fig. 1 shows an illustration.

First, suppose that M is an assignment for I such that $|M| > |M_0|$. We do not require M to be $(4k^2+8k-3)$-close to M_0, but we will actually prove that this is necessary. Observe that M_0 covers each resident except for s_0, so M must cover all residents to satisfy $|M| > |M_0|$. As $A(s_0) = \{b_1\}$, M must assign b_1 to s_0. This implies $M(c(b_1)) = (b'_1, b'_1)$, and therefore we also have $M(s_2(b'_1)) = b_2$, implying $M(c(b_2)) = (b'_2, b'_2)$, and so on. Following this argument, it can be seen that $M(c(b_i)) = (b'_i, b'_i)$ for every $i \in [2k-1]$, and $M(s_2(b'_i)) = b_{i+1}$ for every $i \in [2k-2]$.

We say that a single s *enters* $H^{i,j}$ if $M(s) \in H^{i,j}$ but $M_0(s) \notin H^{i,j}$, and *leaves* $H'^{i,j}$ if $M_0(s) \in H'^{i,j}$ but $M(s) \notin H'^{i,j}$. A couple c *moves* from a hospital h if $M_0(c) = (h, h) \neq M(c)$, and we say that c moves from a set $J \subseteq H$ of hospitals if it moves from a hospital in J. Observe that if c moves from $H^{i,j}$, then two singles leave $H'^{i,j}$, one of them entering $H^{i+1,j}$ if $i \neq k$, and the other entering $H^{i,j+1}$ if $j \neq k$. If a single s leaves $H'^{i,j}$ but does not enter $H^{i+1,j}$ or $H^{i,j+1}$, then $M(s) \in F$ must hold, and therefore there can exist at most $2k$ such single s. Moreover, if a set of m singles enter $H^{i,j}$ then at least $\lceil m/2 \rceil$ couples have to move from $H^{i,j}$. For each $i \in [k]$, exactly one single from $\{s_1(b'_1), s_1(b'_2), \ldots, s_1(b'_k)\}$ enters $H^{1,i}$, and exactly one single from $\{s_1(b'_{k+1}), s_1(b'_{k+2}), \ldots, s_1(b'_{2k-1}), s_2(b'_{2k-1})\}$ enters $H^{i,1}$. These altogether imply that exactly one couple moves from $H^{i,j}$ for each $i, j \in [k]$, and that if s and s' enter $H^{i,j}$ then $M(s) = M(s')$ must hold.

Suppose that c moves from the hospital $h^{i,j}_{x,y}$. Observe that if $j < k$ then a couple must move from $H^{i,j+1}_{x,\bullet}$, and similarly, if $i < k$ then a couple must move from $H^{i+1,j}_{\bullet,y}$. For each $i \in [k]$, letting $\sigma_h(i)$ be x if for some j a couple moves from $H^{i,j}_{x,\bullet}$, and $\sigma_v(i)$ be y if for some j a couple moves from $H^{j,i}_{\bullet,y}$, we obtain that $\sigma_h(i)$ and $\sigma_v(i)$ are well-defined. Observe that by the definition of $H^{i,i}$ we get $\sigma_h(i) = \sigma_v(i) := \sigma(i)$, and from the definition of $H^{i,j}$ we get that if $\sigma(i) = x$ and $\sigma(j) = y$ for some $i \neq j$, then $v_x v_y$ must be an edge in G. Thus, the set $\{v_{\sigma(i)} \mid i \in [k]\}$ forms a clique of size k in G.

Remember that exactly one couple moves from $H^{i,j}$ for each $i, j \in [k]$, which (considering also the size of F) forces exactly two singles to leave $H'^{i,j}$ for each $i, j \in [k]$. Taking into account the couples $c(b_i)$ and the singles $s_1(b'_i), s_2(b'_i)$ for each $i \in [2k-1]$ and the single s_0, we get that M is $4k^2 + 4(2k-1) + 1 = (4k^2 + 8k - 3) = \ell$-close to M_0.

Now, suppose $v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(k)}$ form a clique in G. By defining M as below, it is straightforward to verify that M is an assignment for (S, C, H, f, A) which covers every resident, and is ℓ-close to M_0.

$M(c(b_i)) = (b'_i, b'_i)$ for each $i \in [2k-1]$
$M(c(h^{i,j}_{\sigma(i),\sigma(j)})) = (h'^{i,j}_{\sigma(i),\sigma(j)}, h'^{i,j}_{\sigma(i),\sigma(j)})$ for each $i, j \in [k]$
$M(s_0) = b_1$
$M(s_1(b'_i)) = h^{1,i}_{\sigma(1),\sigma(i)}$ for each $i \in [k]$

$M(s_1(b'_{k+i})) = h^{i,1}_{\sigma(i),\sigma(1)}$ for each $i \in [k-1]$
$M(s_2(b'_{2k-1})) = h^{k,1}_{\sigma(k),\sigma(1)}$
$M(s_2(b'_i)) = b_{i+1}$ for each $i \in [2k-2]$
$M(s_1(h'^{i,j}_{\sigma(i),\sigma(j)})) = h^{i,j+1}_{\sigma(i),\sigma(j+1)}$ for each $i \in [k], j \in [k-1]$
$M(s_2(h'^{i,j}_{\sigma(i),\sigma(j)})) = h^{i+1,j}_{\sigma(i+1),\sigma(j)}$ for each $i \in [k-1], j \in [k]$
$M(s_1(h'^{i,k}_{\sigma(i),\sigma(k)})) = f_i$ for each $i \in [k]$
$M(s_2(h'^{k,i}_{\sigma(k),\sigma(i)})) = f_i$ for each $i \in [k]$
$M(p) = M_0(p)$ for every $p \in S \cup C$ where $M(p)$ was not defined above. $\square$

Let us now remark that the proof of Theorem 7 implicitly contains an FPT reduction from Clique to the decision version of the local search problem for the 2-uniform Maximum Matching with Couples. However, as discussed in Section 2, the presented result is stronger than a $W[1]$-hardness proof.

5. Matching with preferences

In this section, we study the Hospitals/Residents with Couples problem in detail. If no couples are involved, then a stable assignment for a given couples' market with preferences can always be found in linear time with the Gale–Shapley algorithm [1]. In the case when couples are present, a stable assignment may not exist, as first proved by Roth [2]. Here we also give a simple example.

Let $H = \{h_1, h_2, h_3\}$, $S = \emptyset$, $C = \{(a, b), (c, d)\}$ and $f \equiv 1$. The preference lists are defined below. It is straightforward to verify that no stable assignment exists for this cmp which will be denoted by I_0. For example, $M(a) = h_1$, $M(b) = h_2$ and $M(c) = M(d) = \oslash$ is not stable, because (c, d) and (h_1, h_3) form a blocking pair.

$$L((a,b)) : (h_1,h_2), (h_2,h_3), (h_3,h_1) \qquad L(h_1) = L(h_2) = L(h_3) : c, a, b, d$$
$$L((c,d)) : (h_1,h_3), (h_2,h_1), (h_3,h_2).$$

Ronn proved that deciding whether a stable assignment exists for a cmp is NP-complete [13]. As the following example shows, an instance of the Hospitals/Residents with Couples problem may admit stable assignments of different sizes. The example contains a single s, a couple $c = (c_1, c_2)$ and hospitals h_1 and h_2 with capacities $f(h_1) = 2$ and $f(h_2) = 1$. The preference lists are the following:

$$L(s) : h_2, h_1 \qquad L(h_1) : s, c_1, c_2$$
$$L(c) : (h_1, h_1), (\oslash, h_2) \qquad L(h_2) : c_2, s.$$

In this instance, assigning s to h_1 and c to $(\oslash, h_2)$ yields a stable assignment of size 2, whilst assigning s to h_2 and c to (h_1, h_1) results in a stable assignment of size 3. Note that Maximum Hospitals/Residents with Couples problem, where the task is to determine a stable assignment of maximum size for a given cmp, is trivially NP-hard, as it contains the Hospitals/Residents with Couples problem.

The parameterized complexity of Hospitals/Residents with Couples is covered in Section 5.1. In Section 5.2, we present results concerning the applicability of local search for the Maximum Hospitals/Residents with Couples problem.

5.1. Fixed-parameter tractability

The main result of this subsection is Theorem 8, which shows the $W[1]$-hardness of the Hospitals/Residents with Couples problem with parameter $|C|$. As a consequence, the optimization problem Maximum Hospitals/Residents with Couples is also $W[1]$-hard with parameter $|C|$.

However, supposing that a stable assignment has already been determined by some method, it is a valid question whether we can increase its size. Given a cmp I and a stable assignment M_0 for I, the Increase Hospitals/Residents with Couples problem asks for a stable assignment with size at least $|M_0| + 1$. If no couples are involved, then all stable assignments for the instance have the same size, so this problem is trivially polynomial-time solvable. Theorem 8 shows that Increase Hospitals/Residents with Couples is also $W[1]$-hard with parameter $|C|$.

Theorem 8. (1) *The decision version of* Hospitals/Residents with Couples *is* W [1]*-hard with parameter* $|C|$*, even in the* 1*-uniform case.*
(2) *The decision version of* Increase Hospitals/Residents with Couples *is W[1]-hard with parameter* $|C|$*, even in the* 1*-uniform case.*

Before proving Theorem 8, we introduce a special construction that will be very useful in the proof. For a graph G and an integer k, we construct a cmp $I^{G,k} = (S, C, H, f, L)$ as follows. See Fig. 2 for an illustration.

Let $V(G) = \{v_i \mid i \in [n]\}$, $|E(G)| = m$ and let ν be a bijection from $[m]$ into the set $\{(x, y) \mid v_x v_y \in E(G), x < y\}$. First, we construct a *node gadget* $\mathcal{G}^i$ for each $i \in [k]$ and an *edge gadget* $\mathcal{G}^{i,j}$ for each pair $(i, j) \in \binom{[k]}{2}$. The node gadget $\mathcal{G}^i$ contains

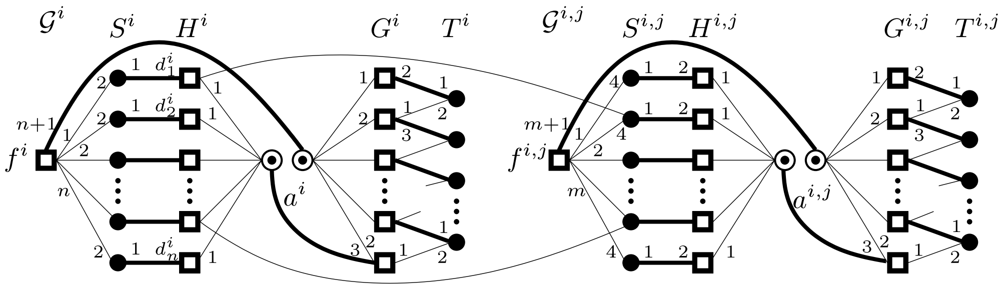

Fig. 2. A node gadget and an edge gadget of $I^{G,k}$. Hospitals are represented by rectangles, singles by black circles, and members of couples by double circles. A hospital h is connected to some resident r if $r \in A^L(h)$. The numbers on the edges represent ranks, bold edges represent $M_0^{G,k}$ from Lemma 9, and d_x^i is for $|Q_x^i| + 2$.

hospitals $H^i \cup G^i \cup \{f^i\}$, singles $S^i \cup T^i$ and a couple a^i. Analogously, the edge gadget $\mathcal{G}^{i,j}$ contains hospitals $H^{i,j} \cup G^{i,j} \cup \{f^{i,j}\}$, singles $S^{i,j} \cup T^{i,j}$ and a couple $a^{i,j}$. Here $T^i = \{t_j^i \mid j \in [n-1]\}$ and $T^{i,j} = \{t_e^{i,j} \mid e \in [m-1]\}$, $H^i = \{h_j^i \mid j \in [n]\}$ and $H^{i,j} = \{h_e^{i,j} \mid e \in [m]\}$, and we define G^i, S^i and $G^{i,j}$, $S^{i,j}$ similarly to H^i and $H^{i,j}$. Observe that $|C| = k + \binom{k}{2}$.

We let $f \equiv 1$, so $I^{G,k}$ is 1-uniform. The precedence lists for each agent in $I^{G,k}$ are defined below. The notation $[X]$ for some set X in a preference list denotes an arbitrary ordering of the elements of X. We write Q_x^i for the set $\{s_e^{i,j} \mid i < j \le k, \exists y : \nu(e) = (x,y)\} \cup \{s_e^{j,i} \mid 1 \le j < i, \exists y : \nu(e) = (y,x)\}$ and $Q_e^{i,j}$ for $\{h_x^i, h_y^j\}$ where $\nu(e) = (x,y)$. The indices in the precedence lists take all possible values if not stated otherwise, and the symbol α can be any index in $[k]$ or a pair of indices in $\binom{[k]}{2}$. If α takes a value in $[k]$ then $N(\alpha) = n$, otherwise $N(\alpha) = m$. (This notation will be used again later on.)

$L(g_x^\alpha) : t_{x-1}^\alpha, a^\alpha(2), t_x^\alpha \quad$ if $1 < x < N(\alpha)$
$L(g_1^\alpha) : a^\alpha(2), t_1^\alpha$
$L(g_{N(\alpha)}^\alpha) : t_{N(\alpha)-1}^\alpha, a^\alpha(2), a^\alpha(1)$
$L(t_x^\alpha) : g_x^\alpha, g_{x+1}^\alpha$
$L(f^\alpha) : s_1^\alpha, s_2^\alpha, \ldots, s_{N(\alpha)}^\alpha, a^\alpha(2)$
$L(a^\alpha) : (g_{N(\alpha)}^\alpha, f^\alpha), (h_1^\alpha, g_{N(\alpha)}^\alpha), (h_2^\alpha, g_{N(\alpha)-1}^\alpha), \ldots, (h_{N(\alpha)}^\alpha, g_1^\alpha)$.

$L(h_x^i) : a^i(1), [Q_x^i], s_x^i$
$L(h_e^{i,j}) : a^{i,j}(1), s_e^{i,j}$
$L(s_x^i) : h_x^i, f^i$
$L(s_e^{i,j}) : h_e^{i,j}, [Q_e^{i,j}], f^{i,j}$

Lemma 9. *For a graph G and $k \in \mathbb{N}$, $I^{G,k}$ has a stable assignment $M_0^{G,k}$ that covers each resident. Moreover, statements* (1)–(3) *are equivalent:*

(1) *There is a clique in G of size k.*
(2) *There is a stable assignment M for $I^{G,k}$ with the following property, which we will call property π: $M(f^{i,j}) \subseteq S^{i,j}$ for each $(i,j) \in \binom{[k]}{2}$.*
(3) *There is a stable assignment for $I^{G,k}$ with property π covering each resident.*

Proof. To see the first claim, we define an assignment M_0 by letting $M_0(a^\alpha) = (g_{N(\alpha)}^\alpha, f^\alpha)$, $M_0(t_x^\alpha) = g_x^\alpha$, and $M_0(s_x^\alpha) = h_x^\alpha$ for all possible values of α and x. As each single and couple is assigned to his or their best choice, M_0 is stable and covers each resident.

To prove (2) $\Rightarrow$ (1), suppose that $I^{G,k}$ has a stable assignment M with property π. Let us define $\sigma(i,j)$ for each $(i,j) \in \binom{[k]}{2}$ such that $M(f^{i,j}) = \{s_{\sigma(i,j)}^{i,j}\}$. Since $s_{\sigma(i,j)}^{i,j}$ prefers $h_{\sigma(i,j)}^{i,j}$ to $f^{i,j}$, $M(h_{\sigma(i,j)}^{i,j}) = \{a^{i,j}(1)\}$ follows from the stability of M. From this, we get that $M(s_e^{i,j}) = h_e^{i,j}$ must hold for each $e \in [m] \setminus \{\sigma(i,j)\}$ since otherwise $s_e^{i,j}$ and $h_e^{i,j}$ would form a blocking pair. Note that each single in $S^{i,j}$ is assigned to a hospital in $H^{i,j} \cup \{f^{i,j}\}$. As this holds for each $(i,j) \in \binom{[k]}{2}$, we get that $M(h_x^i) \subseteq S^i \cup \{a^i(1)\}$ holds for each $i \in [k]$, $x \in [n]$.

Let $\nu(\sigma(i,j)) = (x,y)$ for some $(i,j) \in \binom{[k]}{2}$. Since $s_{\sigma(i,j)}^{i,j}$ prefers the hospitals in $Q_{\sigma(i,j)}^{i,j} = \{h_x^i, h_y^j\}$ to $f^{i,j}$, M can only be stable if both h_x^i and h_y^j prefer their partner in M to $s_{\sigma(i,j)}^{i,j}$. This implies $M(h_x^i) = \{a^i(1)\}$ and $M(h_y^j) = \{a^j(1)\}$. Thus, by defining $\sigma(i)$ to be x if $M(a^i) = (h_x^i, g_{n+1-x}^i)$ for each $i \in [k]$, we obtain $\nu(\sigma(i,j)) = (\sigma(i), \sigma(j))$. From the definition of ν, this implies that $v_{\sigma(i)}$ and $v_{\sigma(j)}$ are adjacent in G. As this holds for every $(i,j) \in \binom{[k]}{2}$, we get that $\{v_{\sigma(i)} \mid i \in [k]\}$ is a clique in G.

Now we prove (1) $\Rightarrow$ (3). If $v_{\sigma(1)}, v_{\sigma(2)}, \ldots, v_{\sigma(k)}$ form a clique in G, then define $\sigma(i,j)$ such that $\sigma(i,j) = \nu^{-1}(\sigma(i), \sigma(j))$. We define a stable assignment M fulfilling property π and covering every resident as follows.

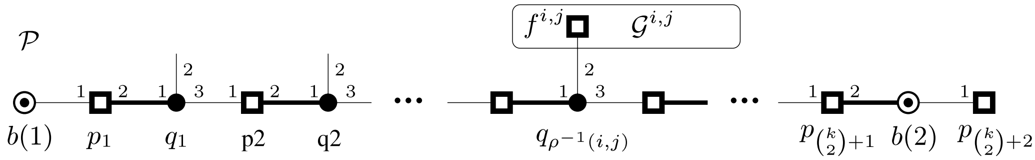

Fig. 3. The path gadget $\mathcal{P}$ in I_2. Bold edges represent M_2.

$M(s^\alpha_{\sigma(\alpha)}) = f^\alpha$
$M(s^\alpha_x) = h^\alpha_x$ if $x \in [N(\alpha)] \setminus \{\sigma(\alpha)\}$
$M(a^\alpha) = (h^\alpha_{\sigma(\alpha)}, g^i_{N(\alpha)+1-\sigma(\alpha)})$
$M(t^\alpha_x) = g^\alpha_x$ if $1 \le x < N(\alpha) + 1 - \sigma(\alpha)$
$M(t^\alpha_x) = g^\alpha_{x+1}$ if $N(\alpha) + 1 - \sigma(\alpha) \le x < N(\alpha)$.

It is not hard to verify the stability of M by simply checking all possibilities to find a blocking pair. (We note that many of the agents are only contained in $I^{G,k}$ to assure that a clique in G indeed implies a stable assignment with the required properties.) As (3) $\Rightarrow$ (2) is trivial, this finishes the proof. $\square$

Proof (of Theorem 8). Let G be an arbitrary graph and $k \in \mathbb{N}$. We construct two 1-uniform cmps I_1 and I_2, together with a stable assignment M_2 for I_2 such that the following three statements are equivalent:

(a) G has a clique of size k,
(b) I_1 has a stable assignment,
(c) I_2 has a stable assignment of size greater than $|M_2|$.

Furthermore, the construction will take FPT time, and there will be $k + 3\binom{k}{2}$ couples in I_1, and $k + \binom{k}{2} + 1$ couples in I_2. Thus, (a) $\iff$ (b) yields an FPT reduction from Clique to Hospitals/Residents with Couples, and (a) $\iff$ (c) yields an FPT reduction from Clique to Increase Hospitals/Residents with Couples.

To get I_1, we simply combine the cmp I_0 having no stable assignment with the cmp $I^{G,k}$. This is done by introducing new couples $b^{i,j}$ and $c^{i,j}$, and new hospitals $\bar{f}^{i,j}_1$ and $\bar{f}^{i,j}_2$ for each $(i,j) \in \binom{[k]}{2}$, and adding these agents to $I^{G,k}$. We preserve the preference lists of $I^{G,k}$, except for hospitals $\left\{f^{i,j} \mid (i,j) \in \binom{[k]}{2}\right\}$, and we give the missing preference lists below.

$L(b^{i,j}) : (f^{i,j}, \bar{f}^{i,j}_1), (\bar{f}^{i,j}_1, \bar{f}^{i,j}_2), (\bar{f}^{i,j}_2, f^{i,j})$
$L(c^{i,j}) : (f^{i,j}, \bar{f}^{i,j}_2), (\bar{f}^{i,j}_1, f^{i,j}), (\bar{f}^{i,j}_2, \bar{f}^{i,j}_1)$
$L(\bar{f}^{i,j}_1) = L(\bar{f}^{i,j}_2) : c^{i,j}(1), b^{i,j}(1), b^{i,j}(2), c^{i,j}(2)$
$L(f^{i,j}) : s^{i,j}_1, s^{i,j}_2, \ldots, s^{i,j}_m, c^{i,j}(1), b^{i,j}(1), b^{i,j}(2), c^{i,j}(2)$.

Observe that if we restrict I_1 to contain only the hospitals $f^{i,j}$, $\bar{f}^{i,j}_1$ and $\bar{f}^{i,j}_2$ and the couples $b^{i,j}$ and $c^{i,j}$ for some $(i,j) \in \binom{[k]}{2}$, we obtain a cmp isomorphic to I_0, having no stable assignment. Therefore, any stable assignment M must assign a single in $S^{i,j}$ to $f^{i,j}$, for each $(i,j) \in \binom{[k]}{2}$. The restriction of such an M on the agents of $I^{G,k}$ must also be stable, because agents of $I^{G,k}$ cannot be assigned by M to agents outside $I^{G,k}$. Thus, by Lemma 9, G has a k-clique.

For the other direction, if there is a k-clique in G, then we can construct a stable assignment M'_1 for I_1 by setting $M'_1(b^{i,j}) = (\bar{f}^{i,j}_1, \bar{f}^{i,j}_2)$, $M'_1(c^{i,j}) = (\oslash, \oslash)$ for each $(i,j) \in \binom{[k]}{2}$, and $M'_1(r) = M^{G,k}_\pi(r)$ for the residents in $I^{G,k}$, where $M_\pi(G,k)$ is the stable assignment for $I^{G,k}$ with property π, guaranteed by Lemma 9. It is easy to see that M'_1 is stable, by using the stability of $M_\pi(G,k)$. This finishes the proof of the first claim.

To construct I_2, we add a *path gadget* $\mathcal{P}$ to $I^{G,k}$ that contains the newly introduced hospitals $\left\{p_i \mid i \in [\binom{k}{2} + 2]\right\}$, singles $\left\{q_i \mid i \in \left[\binom{k}{2}\right]\right\}$ and a couple b. See Fig. 3 for an illustration. As before, we only modify the preferences of the hospitals $\left\{f^{i,j} \mid (i,j) \in \binom{[k]}{2}\right\}$, and we give the missing preference lists below. The notation ρ used there denotes a bijection from $\left[\binom{k}{2}\right]$ into $\binom{[k]}{2}$.

$L(p_1) : b(1), q_1$	$L(p_i) : q_{i-1}, q_i$ if $1 < i \le \binom{k}{2}$
$L(p_{\binom{k}{2}+1}) : q_{\binom{k}{2}}, b(2)$	$L(p_{\binom{k}{2}+2}) : b(2)$
$L(q_i) : p_i, f^{\rho(i)}, p_{i+1}$	$L(f^{i,j}) : s^{i,j}_1, s^{i,j}_2, \ldots, s^{i,j}_m, q_{\rho^{-1}(i,j)}, a^{i,j}(2)$
$L(b) : (\oslash, p_{\binom{k}{2}+1}), (p_1, p_{\binom{k}{2}+2})$.	

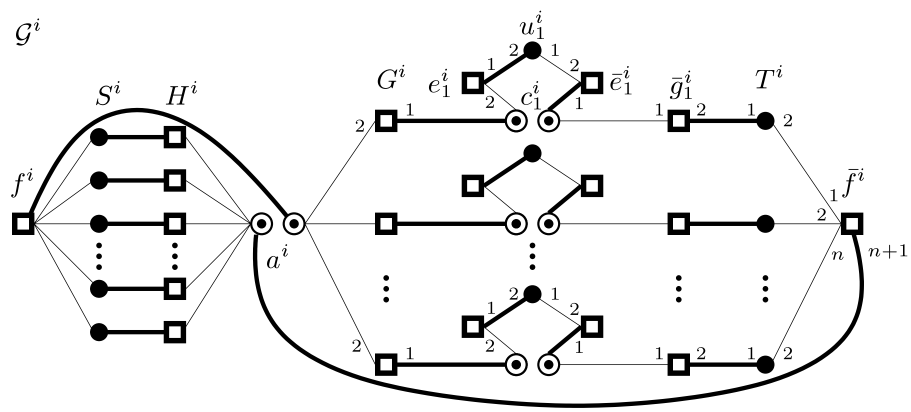

Fig. 4. The modified node gadget in the proof of Theorem 10. Bold edges represent M_3.

We also let $M_2(q_i) = p_i$ for each $i \in \left[\binom{k}{2}\right]$, $M_2(b) = (\emptyset, p_{\binom{k}{2}+1})$, and $M_2(r) = M_0^{G,k}(r)$ for the residents in $I^{G,k}$, where $M_0^{G,k}$ is the stable assignment for $I^{G,k}$, provided by Lemma 9. Note that M_2 is indeed stable.

Suppose, there is a stable assignment M for I_2 with $|M| > |M_2|$. Observe that M_2 covers each resident except for $b(1)$, so M must cover every resident, implying $M(b) = (p_1, p_{\binom{k}{2}+2})$. Also, since $M(h)$ cannot be empty for any hospital h, we get $M(p_i) = \{q_{i-1}\}$ for each $i = \binom{k}{2} + 1, \binom{k}{2}, \ldots, 2$. Therefore, $f^{\rho(i)}$ is beneficial for q_i for each $i \in \left[\binom{k}{2}\right]$, so by the stability of M we obtain $M(f^{i,j}) \subseteq S^{i,j}$ for each $(i,j) \in \binom{[k]}{2}$. Again, the restriction of M on the agents of $I^{G,k}$ must be stable, and so Lemma 9 implies that G has a clique of size k.

Conversely, if there is a k-clique in G, then we can define a stable assignment M_2' for I_2, covering each resident, as follows. We let $M_2'(q_i) = p_{i+1}$ for each $i \in \left[\binom{k}{2}\right]$, $M_2'(b) = (p_1, p_{\binom{k}{2}+2})$, and $M_2'(r) = M_\pi^{G,k}(r)$ for the residents in $I^{G,k}$. Again M_2' is stable, and has size greater than $|M_2|$, proving the second claim. □

5.2. Local search

Here we investigate the applicability of the local search approach for the Maximum Hospitals/Residents with Couples problem. Theorem 10 shows that no permissive local search algorithm is likely to exist for this problem running in FPT time with parameter ℓ, denoting the radius of the explored neighborhood. However, if we regard the combined parameterization $(\ell, |C|)$, then even a strict local search algorithm with FPT running time can be given, as presented in Theorem 11.

Theorem 10. *There is no permissive local search algorithm for the* 1*-uniform* Maximum Hospitals/Residents with Couples *that runs in FPT time with parameter ℓ, unless* W[1] = FPT.

Proof. Let G be a graph and k an integer. First, recall the cmp I_2 defined in the proof of Theorem 8, and observe that the assignment M_2 and the assignment M_2', constructed when a k-clique is present in G, may not be close to each other. Thus, in order to present an FPT-reduction here, we need to modify the node and edge gadgets of I_2. We are going to construct a cmp I_3 together with a stable assignment M_3 for it such that the following statements are equivalent:

(a) G has a clique of size k.
(b) There is a stable assignment for I_3 with size at least $|M_3| + 1$.
(c) There is a stable assignment for I_3 with size at least $|M_3| + 1$ that is ℓ-close to M_3 where $\ell = 8\binom{k}{2} + 7k + 2$.

The construction will take FPT time, hence a permissive local search algorithm for Maximum Hospitals/Residents with Couples that runs in FPT time with parameter ℓ can be used to solve Clique in FPT time.

See Fig. 4 for an illustration of the modifications applied to the instance I_2 in order to get I_3. For each node gadget and edge gadget $\mathcal{G}^\alpha$, we take new singles $\{u_x^\alpha \mid x \in [N(\alpha)]\}$ and the single $t_{N(\alpha)}^\alpha$, new couples $\{c_x^\alpha \mid x \in [N(\alpha)]\}$, and new hospitals $\bigcup_{x\in[N(\alpha)]}\{\bar{g}_x^\alpha, e_x^\alpha, \bar{e}_x^\alpha\} \cup \{\bar{f}^\alpha\}$. For most of the agents we preserve the preferences originally defined for I_2. The modifications and the preference lists of the newly defined agents are as follows.

$L(g_x^\alpha) : c_x^\alpha(1), a^\alpha(2)$
$L(e_x^\alpha) : u_x^\alpha, c_x^\alpha(1)$
$L(\bar{e}_x^\alpha) : c_x^\alpha(2), u_x^\alpha$
$L(\bar{g}_x^\alpha) : c_x^\alpha(2), t_x^\alpha$
$L(a^\alpha) : (\bar{f}^\alpha, f^\alpha), (h_1^\alpha, g_{N(\alpha)}^\alpha), (h_2^\alpha, g_{N(\alpha)-1}^\alpha), \ldots, (h_{N(\alpha)}^\alpha, g_1^\alpha).$
$L(t_x^\alpha) : \bar{g}_x^\alpha, \bar{f}^\alpha$
$L(u_x^\alpha) : \bar{e}_x^\alpha, e_x^\alpha$
$L(c_x^\alpha) = (e_x^\alpha, \bar{g}_x^\alpha), (g_x^\alpha, \bar{e}_x^\alpha)$
$L(\bar{f}^\alpha) : t_1^\alpha, t_2^\alpha, \ldots, t_{N(\alpha)}^\alpha, a^\alpha(1)$

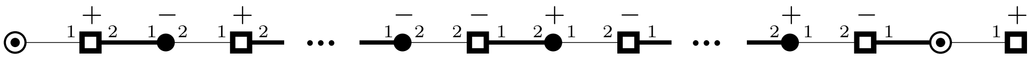

Fig. 5. A possible component of G^δ. Winners and losers are marked by '+' and '−' signs, respectively. Bold edges represent M_0, normal edges represent M.

We also define $M_3(a^\alpha) = (\bar{f}^\alpha, f^\alpha)$, $M_3(c_x^\alpha) = (g_x^\alpha, \bar{e}_x^\alpha)$, $M_3(u_x^\alpha) = e_x^\alpha$ and $M_3(t_x^\alpha) = \bar{g}_x^\alpha$ for all possible values of α and x, and for each remaining resident r let $M_3(r) = M_2(r)$. It is easy to observe that M_3 is stable, and covers each resident except for $b(1)$.

Supposing that there is a stable assignment M with size greater than $|M_3|$ and using exactly the same arguments as in the proof of Theorem 8, we get $M(b) = (p_1, p_{\binom{k}{2}+2})$, $M(q_i) = (p_{i+1})$ for each $i \in \left[\binom{k}{2}\right]$, and $M(f^{i,j}) \subseteq S^{i,j}$ for each $(i,j) \in \binom{[k]}{2}$. By following the argument proving (2) ⇒ (1) in Lemma 9, we again obtain that G must have a k-clique. (The modifications of the gadgets in I_3 to do not affect that reasoning.) This proves (b) ⇒ (a).

Clearly, (c) ⇒ (b) is trivial, so we only have to prove (a) ⇒ (c). Suppose that G has a clique $\{v_{\sigma(i)} \mid i \in [k]\}$. We again let $\sigma(i,j) = \nu^{-1}(\sigma(i), \sigma(j))$, and we write $\sigma'(\alpha)$ for $N(\alpha) + 1 - \sigma(\alpha)$. We define a stable assignment M_3' for I in a very similar fashion as in the previous proofs:

$$\begin{array}{ll}
M_3'(b) = (p_1, p_{\binom{k}{2}+2}) & M_3'(u^\alpha_{\sigma'(\alpha)}) = \bar{e}^\alpha_{\sigma'(\alpha)} \\
M_3'(q_i) = p_{i+1} \quad \text{for each } i \in \left[\binom{k}{2}\right] & M_3'(s^\alpha_{\sigma(\alpha)}) = f^\alpha \\
M_3'(a^\alpha) = (h^\alpha_{\sigma(\alpha)}, g^\alpha_{\sigma'(\alpha)}) & M_3'(t^\alpha_{\sigma'(\alpha)}) = \bar{f}^\alpha \\
M_3'(c^\alpha_{\sigma'(\alpha)}) = (e^\alpha_{\sigma'(\alpha)}, \bar{g}^\alpha_{\sigma'(\alpha)}). &
\end{array}$$

For each remaining resident r we let $M_3'(r) = M_3(r)$. It is straightforward to verify that M_3' is stable, and it is ℓ-close to M_0. □

Before stating our last result, we describe the trick of cloning hospitals, already mentioned in Section 4, for the case involving preferences. For each hospital $h \in H$ in a given cmp, we take $f(h)$ copies of h by replacing h with new hospitals $h^1, \ldots, h^{f(h)}$, each having capacity 1. The preference lists of these hospitals agree with the original preference list of h. For each single s containing h in its preference list, we replace h in the list $L(s)$ by the series $h^1, \ldots, h^{f(h)}$. For a couple c containing a pair (h, g) of two hospitals in $L(c)$, we replace (h, g) by a series formed by the elements of $\{(h^i, g^j) : i \in [f(h)], j \in [f(g)]\}$ such that (h^i, g^j) precedes $(h^{i'}, g^{j'})$ if $i < i'$, or $i = i'$ and $j < j'$. (We assume that the cases $h = \emptyset$ and $g = \emptyset$ are also clear.)

Now, if M is an assignment for the original cmp I, then it defines an assignment M^c for the cmp I^c obtained by the above cloning process, as follows. If M assigns r to h and there are $i - 1$ residents in $M(h)$ that h prefers to r, then let $M^c(r) = h^i$. If $M(r) = \emptyset$ for some r, then we let $M^c(r) = \emptyset$ as well. Observe that if M is stable then M^c is also stable. Conversely, it is not hard to see that a stable assignment for I^c can be transformed in the straightforward way to a stable assignment for I.

Theorem 11. *There is a strict local search algorithm for* Maximum Hospitals/Residents with Couples *running in FPT time with combined parameter* $(\ell, |C|)$.

Proof. Let $I = (S, C, H, f, L)$ be given with the stable assignment M_0 and the integer ℓ. W.l.o.g. we may assume that $f \equiv 1$, as otherwise we can apply the trick of cloning the hospitals, as argued above. Thus, if $M(r) = h$ for some resident r, then we will write $M(h) = r$ instead of $M(h) = \{r\}$.

Before describing the strict local search algorithm for Maximum Hospitals/Residents with Couples, we introduce some notation to capture the structure of the solution. The bipartite graph G underlying I has vertex set $H \cup R$ and edge set $E = \{hr \mid h \in H, r \in A^L(h)\}$. Clearly, an assignment M for I determines a matching $E(M)$ in G in the natural way: $hr \in E(M)$ if and only if $M(r) = h$ for some resident r and hospital h. Suppose that M is a closest solution, i.e. a stable assignment for I with $|M| > |M_0|$ and $d(M, M_0) \leq \ell$ that is the closest to M_0 among all such assignments. Let $A^\delta = \{a \in R \cup H \mid M(a) \neq M_0(a)\}$, and E^δ be the symmetric difference of $E(M_0)$ and $E(M)$. Note that E^δ covers exactly the vertices of A^δ, and $G^\delta = (A^\delta, E^\delta)$ is the union of paths and cycles which contain edges from M_0 and M in an alternating manner. It is well-known that for a cmp not containing couples, every stable assignment covers exactly the same agents [30]. Thus, it is easy to see that the stability of M and M_0 imply that if a component of G^δ does not contain any resident from $R \setminus S$, then it must be a cycle. Let $\mathcal{K}_0$ denote the set of such cycles, and $\mathcal{K}_1$ the set of the remaining components of G^δ. We write C^δ for $(R \setminus S) \cap A^\delta$, and we define $B(a) = \{b \mid a \text{ is beneficial for } b \text{ w.r.t. } M_0\}$ for every $a \in S \cup H$. We also let $S^+ = \{s \in S \mid M(s) \text{ is beneficial for } s \text{ w.r.t. } M_0\}$, and $S^- = \{s \in S \mid M_0(s) \text{ is beneficial for } s \text{ w.r.t. } M\}$. Note that $S^+ \cup S^- = S \cap A^\delta$. We define H^+ and H^- analogously. We call agents in $A^+ = S^+ \cup H^+$ *winners* and agents in $A^- = S^- \cup H^-$ *losers*. For a simple illustration see Fig. 5.

Now, we describe an algorithm that finds vertices of A^δ. The algorithm first branches on guessing $|A^\delta|$ and a copy $\bar{G}$ of the graph G^δ. Let φ denote an isomorphism from $\bar{G}$ to G^δ. The algorithm also guesses the vertex sets $\varphi^{-1}(C^\delta)$, $\varphi^{-1}(H^+)$, $\varphi^{-1}(H^-)$, $\varphi^{-1}(S^+)$, $\varphi^{-1}(S^-)$, and edge sets $\bar{E}_{M_0}$ and $\bar{E}_M$ denoting $\varphi^{-1}(E(M_0) \cap E^\delta)$ and $\varphi^{-1}(E(M) \cap E^\delta)$, respectively. Since $|A^\delta| \leq 2\ell$, it can be achieved by careful implementation that the algorithm branches into at most $(2\ell)6^{2\ell}$ directions in this phase.

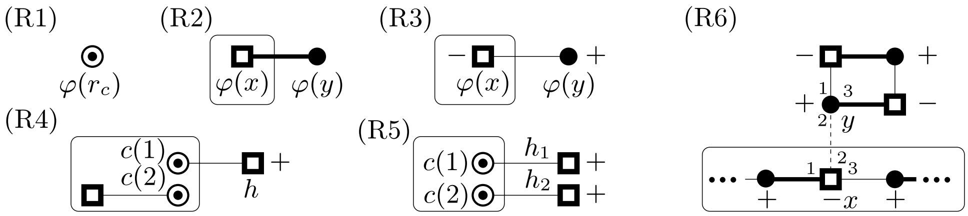

Fig. 6. Figure (Ri) shows a subgraph of G^δ illustrating Extension Rule i, for $i \in \{1, 2, \dots, 6\}$. We represent agents of $\varphi(X)$ by enclosing them in a rectangular box. Bold edges represent $\bar{E}_{M_0}$ and normal edges represent $\bar{E}_M$.

Next, we apply the technique of color-coding [31], in order to help the localization of A^δ. To this end, the algorithm colors the vertices of G with $|A^\delta| \le 2\ell$ colors randomly with uniform and independent distribution; $\gamma(a)$ denotes the color of a. The coloring γ is *nice*, if $\gamma(\varphi(a)) = \Gamma(a)$ for each $a \in V(\bar{G})$, where Γ is an arbitrary fixed ordering of $V(\bar{G})$, i.e. a bijection from $V(\bar{G})$ to $[|A^\delta|]$. From now on, we suppose that γ is nice, which clearly holds with probability $|A^\delta|^{-|A^\delta|} \ge (2\ell)^{-2\ell}$.

Given a coloring, the algorithm grows a subset $X \subseteq V(\bar{G})$ on which φ is already known. It applies the following extension rules repeatedly, until none of them is applicable. When Extension Rule 1 is applied, the algorithm branches into at most $2|C|$ branches, but no other branchings are involved. We write $\bar{X} = V(\bar{G}) \setminus X$. See Fig. 6 for an illustration.

Extension Rule 1 (*Guessing a Member of a Couple*). Applicable if $c \in \bar{X} \cap \varphi^{-1}(C^\delta)$. In this case we simply branch on the vertices of $(R \setminus S) \cap \{a \mid \gamma(a) = \Gamma(c)\}$ to choose $\varphi(c)$. Note that this means at most $2|C|$ branches.

Extension Rule 2 (*Finding Pairs by M_0*). Applicable if $x \in X, y \in \bar{X}$ and $xy \in \bar{E}_{M_0}$ for some x and y. Clearly, we get $\varphi(y) = M_0(\varphi(x))$, so we can extend φ by adding y to X.

Extension Rule 3 (*Finding Pairs by M for Losers*). Applicable if $x \in X \cap \varphi^{-1}(A^-)$, $y \in \bar{X} \cap \varphi^{-1}(A^+)$ and $xy \in \bar{E}_M$ for some x and y. Let y^* be the first element in the preference list $L(\varphi(x))$ contained in the set $B(\varphi(x))$ having color $\Gamma(y)$. We claim $y^* = \varphi(y)$. Clearly, $\varphi(y) \in B(\varphi(x))$ holds because $\varphi(y)$ is a winner, and its color must be $\Gamma(y)$ as γ is nice. Now, suppose for contradiction that y^* precedes $\varphi(y)$ in $L(\varphi(x))$. Since the only vertex in A^δ having color $\Gamma(y)$ is $\varphi(y)$, we get $M(y^*) = M_0(y^*)$ implying that y^* and $\varphi(x)$ form a blocking pair for M. Thus, $\varphi(y) = y^*$ can be found in linear time, so we can extend φ by adding y to X.

Extension Rule 4 (*Finding Pairs by M for Couples with One Winner Hospital*). Applicable if $c(i) \in C^\delta \cap \varphi(X)$, $y \in \varphi^{-1}(H^+) \cap \bar{X}$, $\varphi^{-1}(c(i))y \in \bar{E}_M$, and $M(c(i'))$ is already known for some $c \in C$, $i \ne i'$ and y. W.l.o.g. we assume $i = 1$. Let h be defined such that $(h, M(c(2)))$ is the first element in $L(c)$ for which $h \in B(c(1))$ and h has color $\Gamma(y)$. We claim $\varphi(y) = h$. Observe that $\varphi(y) \in B(c(1))$ must hold because $\varphi(y)$ is a winner. As γ is nice, $\varphi(y)$ indeed has color $\Gamma(y)$. Thus, if $h \ne \varphi(y)$ then $(h, M(c(2)))$ precedes $(\varphi(y), M(c(2)))$ in $L(c)$, but this implies that the couple c and $(h, M(c(2)))$ form a blocking pair for M. Therefore we get $\varphi(y) = h$, and we can extend φ in linear time by adding y to X.

Extension Rule 5 (*Finding Pairs by M for Couples with Two Winner Hospitals*). Applicable if $c(i) \in C^\delta \cap \varphi(X)$, $y_i \in \varphi^{-1}(H^+) \cap \bar{X}$, and $\varphi^{-1}(c(i))y_i \in \bar{E}_M$ holds for both $i \in \{1, 2\}$, for some $c \in C$, y_1 and y_2. We let (h_1, h_2) be the first element in $L(c)$ such that $h_i \in B(c(i))$ and $\gamma(h_i) = \Gamma(y_i)$ for both $i \in \{1, 2\}$. Using the same arguments as in the previous case, we can argue that $\varphi(y_1) = h_1$ and $\varphi(y_2) = h_2$ hold. Thus, in this case we can extend φ in linear time by adding both y_1 and y_2 to X.

Extension Rule 6 (*Dissolving a Blocking Pair*). Applicable if $M(a) \in \varphi(X)$ if and only if $a \in \varphi(X)$ for all $a \in A^\delta$, and xy is a blocking pair for the *actual assignment* M_X. We define M_X by setting $M_X(a) = M_0(a)$ if $a \notin \varphi(X)$ and $M_X(a) = M(a)$ if $a \in \varphi(X)$, for each agent a. Note that by our first condition, M_X is indeed an assignment. Now, as xy cannot be a blocking pair for M or M_0, either $x \in \varphi(X)$ and $y \in A^\delta \setminus \varphi(X)$, or vice versa. W.l.o.g. we suppose the former. By defining $\bar{y} \in V(\bar{G})$ such that $\Gamma(\bar{y}) = \gamma(y)$, it can be seen that $\varphi(\bar{y}) = y$ must hold because γ is nice. Thus, φ can be extended by adding $\bar{y}$ to X.

Lemma 12. *If none of the extension rules is applicable, then* $\varphi(X) = A^\delta$.

Proof. First, $\varphi(X) \supseteq C^\delta$ is trivial, as Extension Rule 1 is not applicable.

Claim 1. $\varphi(X) \supseteq (H^- \cup S^+) \cap V(\mathcal{K}_1)$.

Suppose $a \in (H^- \cup S^+) \cap V(\mathcal{K}_1) \setminus \varphi(X)$ is chosen such that the distance $d^C(a)$ is minimal, where $d^C(a)$ is the minimum length of a path P in G^δ from a to some $c \in C^\delta$ such that the first edge of P is in $E(M_0)$ if $a \in H$ and it is in $E(M)$ if $a \in S$. If no such path exists then let $d^C(a) = \infty$.

First, if a is a winner single, then $M(a) \ne \emptyset$, and since a and $M(a)$ cannot be a blocking pair for M_0, $M(a)$ must be a loser hospital. Now, if $M(a) \in \varphi(X)$ then Extension Rule 3 is applicable, a contradiction. Thus $M(a) \notin \varphi(X)$, but as $M(a)$ is on the path defining $d^C(a)$, we get $d^C(M(a)) < d^C(a)$ contradicting to the choice of a. (Note that $d^C(a) \ne \infty$ as $a \in V(\mathcal{K}_1)$.) Second, if a is a loser hospital, then $M_0(a) \ne \emptyset$. Observe that if $M_0(a) \in \varphi(X)$ then Extension Rule 2 is applicable, which cannot be the case, so $M_0(a)$ can only be a single in $S \setminus \varphi(X)$. If $M_0(a)$ were a loser, then a and $M_0(a)$ would form a blocking pair for M, so we obtain $M_0(a) \in S^+ \setminus \varphi(X)$. But this implies $d^C(M_0(a)) < d^C(a)$, a contradiction. Thus, $\varphi(X)$ indeed contains $(H^- \cup S^+) \cap V(\mathcal{K}_1)$.

Claim 2. $\varphi(X) \supseteq V(\mathcal{K}_1)$.

By the statement of Claim 1, we only have to prove that $(H^+ \cup S^-) \cap V(\mathcal{K}_1) \setminus \varphi(X)$ is empty. Analogously as in Claim 1, we choose $a \in (H^+ \cup S^-) \cap V(\mathcal{K}_1) \setminus \varphi(X)$ such that the distance $d'^C(a)$ is minimal, where $d'^C(a)$ is the minimum length of a path P in G^δ from a to some $c \in C^\delta$ such that the first edge of P is in $E(M)$ if $a \in H$ and it is in $E(M_0)$ if $a \in S$. If no such path exists then let $d'^C(a) = \infty$. Note that $d'^C \neq d^C$, as the requirements for the first edge of the path P are different.

First, if a is a loser single, then $M_0(a) \neq \emptyset$, and since a and $M_0(a)$ cannot be a blocking pair for M, $M_0(a)$ must be a winner hospital. Now, if $M_0(a) \in \varphi(X)$ then Extension Rule 2 is applicable, a contradiction. Thus $M_0(a) \notin \varphi(X)$, but as $M_0(a)$ is on the path defining $d'^C(a)$, we get $d'^C(M_0(a)) < d'^C(a)$ contradicting to the choice of a. Again, $d'^C(a) \neq \infty$ as $a \in V(\mathcal{K}_1)$.

Second, if a is a winner hospital, then $M(a) \neq \emptyset$. Observe that if $M(a)$ is a member of some couple c, then if $M(c(i))$ is not known for some $i \in \{1, 2\}$, then $M(c(i))$ can only be a winner hospital by Claim 1, so Extension Rules 4 and 5 is applicable. If $M(a)$ were a winner single, then a and $M(a)$ would form a blocking pair for M_0, so we obtain $M(a) \in S^-$. Now, if $M(a) \in S^- \cap \varphi(X)$ then Extension Rule 3 is applicable. Thus, only $M(a) \in S^- \setminus \varphi(X)$ is possible. But this implies $d'^C(M(a)) < d'^C(a)$, which is a contradiction proving Claim 2.

Claim 3. $\varphi(X) \supseteq V(\mathcal{K}_0)$.

As already mentioned, each component of $\mathcal{K}_0$ is a cycle, and it easy to see that it must contain vertices from A^+ and A^- in an alternating manner. Thus, if neither of Extension Rules 2 and 3 is applicable, then each component of $\mathcal{K}_0$ is totally contained in either $A^\delta \setminus \varphi(X)$ or in $\varphi(X)$. Thus, the first condition of Extension Rule 6 must hold. Now, if $\varphi(X) \neq A^\delta$ then clearly $M_X \neq M$. As M_X is closer to M_0 than M, and M is a closest solution, M_X cannot be stable. Thus Extension Rule 6 is applicable, a contradiction.

Now, Claims 1–3 together imply the lemma. □

If no extension rule is applicable, then we can easily obtain the solution M by Lemma 12. Each step takes linear time, the number of steps is at most 2ℓ, and the algorithm branches into at most $(2\ell)6^{2\ell}(2|C|)^\ell$ branches in total, thus the overall running time is $O(\ell(72|C|)^\ell|I|)$. The output is correct if the coloring γ is nice, which holds with probability at least $(2\ell)^{-2\ell}$. To derandomize the algorithm, we can use the standard method of k-perfect hash functions [31] instead of randomly coloring the vertices of G. This yields a running time of $O(\ell^{O(\ell)}|C|^\ell|I| \log |I|)$. □

6. Summary

We addressed the parameterized complexity of different assignment problems in models where couples can be present in the market, considering them also in the context of local search.

First, we investigated the extension of standard matching problems to the case where couples are involved. We obtained a randomized fixed-parameter tractable algorithm for the Maximum Matching with Couples problem in the case where the parameter is the number of couples (Theorem 1). We applied the presented algorithm for a problem arising in the area of scheduling, where the task is to find a minimum makespan scheduling of jobs with processing restrictions, assuming that the job length are in $\{1, p\}$ for some integer p (Theorem 6).

We also examined the applicability of local search algorithms for Maximum Matching with Couples, and we obtained that no permissive algorithm can run in FPT time if the parameter is the radius of the explored neighborhood, even if all hospitals have capacity 2, unless W[1] = FPT (Theorem 7).

Next, we studied the parameterized complexity of stable assignment problems, modeling situations where the agents of the market have preferences and may form couples. We obtained that the Hospitals/Residents with Couples problem is $W[1]$-hard, if the parameter is the number of couples (Theorem 8). On the one hand, we showed that no permissive algorithm for Hospitals/Residents with Couples runs in FPT time if the parameter is the radius of the explored neighborhood, even if all hospitals have capacity 1, unless W[1] = FPT (Theorem 10). On the other hand, we presented a strict local search algorithm for this problem, if both the radius of the explored neighborhood and also the number of couples are parameters (Theorem 11).

Acknowledgements

We would like to thank David Manlove, Péter Biró, and Rolf Niedermeier for their useful advice and comments.

References

[1] D. Gale, L.S. Shapley, College admissions and the stability of marriage, Amer. Math. Monthly 69 (1) (1962) 9–15.
[2] A.E. Roth, The evolution of the labor market for medical interns and residents: a case study in game theory, J. Political Econom. 92 (1984) 991–1016.
[3] A.E. Roth, M. Sotomayor, Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis, in: Econometric Society Monographs, vol. 18, Cambridge University Press, Cambridge, MA, 1990.
[4] P.A. Robards, Applying Two-Sided Matching Processes to the United States Navy Enlisted Assignment Process, Ph.D. Thesis, Naval Postgraduate School Monterey CA, 2001.
[5] D.F. Manlove, Stable marriage with ties and unacceptable partners, Technical Report, University of Glasgow, Department of Computing Science, 1999.

[6] D.F. Manlove, R.W. Irving, K. Iwama, S. Miyazaki, Y. Morita, Hard variants of stable marriage, Theor. Comput. Sci. 276 (1–2) (2002) 261–279.
[7] R.W. Irving, An efficient algorithm for the stable roommates problem, J. Algorithms 6 (4) (1985) 577–595.
[8] R.W. Irving, D.F. Manlove, The stable roommates problem with ties, J. Algorithms 43 (1) (2002) 85–105.
[9] A.E. Roth, A natural experiment in the organization of entry-level labor markets: regional markets for new physicians and surgeons in the United Kingdom, American Economic Review 81 (3) (1991) 415–440.
[10] C. Blair, The lattice structure of the set of stable matchings with multiple partners, Math. Oper. Res. 13 (4) (1988) 619–628.
[11] F. Echenique, J. Oviedo, A theory of stability in many-to-many matching markets, Theor. Econ. 1 (2) (2006) 233–273.
[12] A.E. Roth, E. Peranson, The redesign of the matching market for American physicians: some engineering aspects of economic design, American Economic Review 89 (4) (1999) 748–780.
[13] E. Ronn, NP-complete stable matching problems, J. Algorithms 11 (2) (1990) 285–304.
[14] B. Klaus, F. Klijn, Stable matchings and preferences of couples, J. Econom. Theory 121 (1) (2005) 75–106.
[15] D. Cantala, Matching markets: the particular case of couples, Economics Bulletin 3 (45) (2004) 1–11.
[16] B. Dutta, J. Massó, Stability of matchings when individuals have preferences over colleagues, J. Econom. Theory 75 (2) (1997) 464–475.
[17] E.J. McDermid, D.F. Manlove, Keeping partners together: algorithmic results for the hospitals/residents problem with couples, J. Combin. Optim. 19 (3) (2010) 279–303.
[18] R.G. Downey, M.R. Fellows, Parameterized Complexity, in: Monographs in Computer Science, Springer-Verlag, New York, 1999.
[19] E.H.L. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Princeton University Press, Princeton, NJ, 2003, Reprint of the 1997 original [Wiley, Chichester].
[20] D. Marx, Local search, in: Parameterized Complexity News, vol. 3, 2008, pp. 7–8.
[21] S. Khuller, R. Bhatia, R. Pless, On local search and placement of meters in networks, in: SODA'00: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000, pp. 319–328.
[22] A. Krokhin, D. Marx, On the hardness of losing weight, in: ICALP 2008: 35rd International Colloquium on Automata, Languages and Programming, in: Lecture Notes in Computer Science, vol. 5125, Springer, Berlin, 2008, pp. 662–673.
[23] D. Marx, Searching the k-change neighborhood for TSP is W[1]-hard, Oper. Res. Lett. 36 (1) (2008) 31–36.
[24] D. Marx, A parameterized view on matroid optimization problems, Theor. Comput. Sci. 410 (44) (2009) 4471–4479.
[25] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, in: Oxford Lecture Series in Mathematics and its Applications, vol. 31, Oxford University Press, Oxford, 2006.
[26] J. Flum, M. Grohe, Parameterized Complexity Theory, in: Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, New York, 2006.
[27] C.A. Glass, H. Kellerer, Parallel machine scheduling with job assignment restrictions, Nav. Res. Logist. 54 (3) (2007) 250–257.
[28] P. Biró, E.J. McDermid, Matching with sizes (or scheduling with processing set restrictions), Technical Report TR-2010-307, University of Glasgow, Department of Computing Science, 2010.
[29] J.Y.-T. Leung, C.-L. Li, Scheduling with processing set restrictions: a survey, Int. J. Prod. Econ. 116 (2008) 251–262.
[30] D. Gale, M. Sotomayor, Some remarks on the stable matching problem, Discrete Appl. Math. 11 (3) (1985) 223–232.
[31] N. Alon, R. Yuster, U. Zwick, Color-coding, J. ACM 42 (4) (1995) 844–856.